Proposing a Pre-processing Optimisation applied to the Classical Integer Programming Model for Statistical Disclosure Control Draft 12

Martin Serpell, Alistair Clark, Jim Smith and Andrea Staggemeier

Martin Serpell, Alistair Clark and Jim Smith
University of the West of England,
Bristol, United Kingdom
Martin2.Serpell@uwe.ac.uk, Alistair.Clark@uwe.ac.uk and James.Smith@uwe.ac.uk

Andrea Staggemeier
Head of Statistical Tools and Operational Research Team
Office for National Statistics
Newport, United Kingdom
Andrea.Staggemeier@ons.gsi.gov.uk

Abstract. A pre-processing optimisation is proposed that can be applied to the integer and mixed integer linear programming models that are used to solve the cell suppression problem in statistical disclosure control. In this paper we report our initial findings and acknowledge that there is much more work to be done. Early indications are that the pre-processing optimisation will considerably reduce the resources required by the solver hence allowing either statistical tables to be protected quicker or larger statistical tables to be protected. This pre-processing optimisation may be suitable for application to the τ-Argus Optimal Method used in protecting statistical tables.

Key words: Statistical Disclosure Control, Cell Suppression Problem, Classical Model, Pre-processing Optimisation, External Attacker.

1 Introduction

Many statistical tables are published with some of the table cells suppressed (left blank). This is done to prevent the disclosure of individual respondents which contributed to the cell value. Cells that failed the primary rule are called primary, or sensitive, cells and must be protected by additional suppressed cells called secondary cells. Choosing which secondary cells to suppress is known, in the literature, as the cell suppression problem. The cell suppression problem involves choosing a set of secondary cells that will remove the risk of disclosing the values of the primary cells whilst also minimising the information loss from the published statistical table.

The cell suppression problem is a member of the class of NP-hard problems when solving for optimality. In fact, the problem of finding a secondary suppression pattern is easy to be achieved, for example if all cells are suppressed this is a feasible pattern but clearly not optimal. It is when solving the cell suppression problem optimally that as the size of the table to be protected grows the number of possible solutions that need to be evaluated grows much quicker. For a table with n cells there are 2^n possible suppression patterns. This means that when trying to find an optimal solution the computational time required grows rapidly as the table size grows, making finding optimal solutions for large tables difficult. Because the cell suppression problem is NP-hard MIP techniques can only find the optimal solution for small and medium sized statistical tables.

It is known that removing anything that is redundant from the mathematical program can make an efficiency gain. For example redundant equations, variables and protection levels can be removed. Another pre-processing efficiency gain can be obtained by removing any table cells that have the value set to zero or whose values must be published, subject to adjusting any marginal totals necessary. This decreases the number of working variables and constraints that the solver requires to find a solution, which in turn allows larger statistical tables to be protected.

Linear programming models and local search algorithms are used on relaxed cell suppression problems to obtain near optimal solutions when integer programming models are infeasible. Some have moved away from trying to calculate the optimal solution and have instead employed heuristic techniques to find near optimal solutions quickly. Others have employed hybrid algorithms that combine linear programming and heuristic techniques [2] [8].

This paper will present further improvements which are obtained when looking at the inferences made by an external attacker to a table. Section 2 presents definitions to the problem. Section 3 puts forward a proposition for a pre-processing optimisation. Section 4 describes how the pre-processing optimisation can be implemented. Section 5 applies the pre-processing optimisation to the classical IP model for SDC. Section 6 describes our experimental setup. Section 7 contains our results. Section 8 contains our preliminary conclusions and section 9 lists further research.

2 Definitions

The *external attacker* wishes to deduce the values of cells that have been suppressed in a published statistical table, in order to glean confidential information. The assumption made in the literature is that the external attacker has only the knowledge which is provided in the published table, i.e. he is not aware which suppressed cells are primary nor secondary but he knows that there is a number of suppressed cells in the table and their location (disclosure pattern). As each table has row and column totals, often referred to as marginals, the *external attacker* is able to calculate lower and upper bounds, feasibility range, for each of the suppressed cells by solving a set of linear constraint equations [1] [6].

A statistical table with marginal totals can be represented as a set of cells, please see details of the model in [1] and [6], $a_i, i = 1, ..., n$, satisfying m linear constraint equations such that $Ma = 0$, where M_{ij} has one of the values $\{0, +1, -1\}$.

$$\sum_{i=1}^{n} M_{ij} a_i = 0, j = 1, ..., m$$

The statistical agency will define a set P of primary cells whose publication will be suppressed in order to protect the confidentiality of the contributors to those cells. The statistical agency will provide lower and upper protection levels (lpl and upl) for each cell in P such that an external attacker must not be able to calculate a_p within the range lpl_p to upl_p. For a_p to be safe

$$\underline{a_p} \leq lpl_p \quad \text{and} \quad \overline{a_p} \geq upl_p$$

where $\underline{a_p}$ is the lower bound and $\overline{a_p}$ the upper bound of the feasible range that the external attacker can calculate for a_p if only the primary cells P have been suppressed [1].

$$\begin{array}{lll} \underline{a_p} = min\ x_p & & \overline{a_p} = max\ x_p \\ \quad s.t.\ Mx = 0 & \text{and} & \quad s.t.\ Mx = 0 \\ \quad\quad x_i \geq 0,\ i \in P & & \quad\quad x_i \geq 0,\ i \in P \\ \quad\quad x_i = a_i, i \notin P & & \quad\quad x_i = a_i, i \notin P \end{array}$$

If the external attacker is able to calculate $\underline{a_p} > lpl_p$ or $\overline{a_p} < upl_p$ then a_p is unsafe (a_p can be disclosed). It should be noted that we are considering the external attacker on tables which have not yet been protected by secondary suppressed cells in order to gauge the level of disclosiveness of the tables for our pre-processing optimisation.

0	lpl_p	a_p	upl_p	∞
$\underline{a_p} \leq lpl_p$	$lpl_p < \underline{a_p}$	$\overline{a_p} < upl_p$	$upl_p \leq \overline{a_p}$	
a_p is safe	a_p is unsafe	a_p is unsafe	a_p is safe	

Noting that some primary cells may occur alone in a marginal total, whereas others (e.g. those sharing rows/columns) may effectively protect each other, we define the following partition of the set of primary cells P.

An *exposed* primary cell in a statistical table with marginal totals is one whose value can be calculated, within a given lower and upper protection limit, by an external attacker when only the primary cells P have been suppressed. That is to say, p is a member of the set E of *exposed* primary cells if $\underline{a_p} > lpl_p$ or $\overline{a_p} < upl_p$. $E \subseteq P$.

A *not exposed* primary cell in a statistical table with marginal totals is one whose value cannot be calculated, within a given lower and upper protection

limit, by an external attacker when only the primary cells P have been suppressed. That is to say, p is a member of the set N of *not exposed* primary cells if $\underline{a_p} \leq lpl_p$ and $\overline{a_p} \geq upl_p$. $N \subseteq P$, $E \cup N = P$ and $E \cap N = \{\}$. The reason why there are not exposed primary cells in a statistical table is due to their locations in that table. Each not exposed primary cell receives sufficient protection from other primary cells in the table to prevent an external attacker from being able to calculate a feasible range of values within the given protection level.

Proposition 1: As *not exposed* primary cells are already sufficiently protected they do not require secondary cells for their protection.

An *initially exposed* primary cell is a primary cell that can be exposed, by an external attacker when only the primary cells P have been suppressed, without requiring the exposure of any other primary cell. For example there may be only one primary cell in a row or column. Let L_p be the subset of linear equations M that contain the value $+1$ or -1 in the locations for a_p, $L_p \subseteq M$. This subset L_p only contains the linear equations that apply to a_p. Then we can say that p is a member of the set I of *initially exposed* primary cells if $\underline{a_p} > lpl_p$ or $\overline{a_p} < upl_p$, when,

$$\begin{array}{lll} \underline{a_p} = min\ x_p & & \overline{a_p} = max\ x_p \\ \quad s.t.\ L_p x = 0 & \text{and} & \quad s.t.\ L_p x = 0 \\ \quad\quad x_i \geq 0,\ i \in P & & \quad\quad x_i \geq 0,\ i \in P \\ \quad\quad x_i = a_i, i \notin P & & \quad\quad x_i = a_i, i \notin P \end{array}$$

$I \subseteq E$.

Conversely we can say that p is not a member of I if $\underline{a_p} \leq lpl_p$ and $\overline{a_p} \geq upl_p$.

A *consequentially exposed* primary cell is an *exposed* primary cell that is not an *initially exposed* primary cell. That is to say, p is a member of the set C of *consequentially exposed* primary cells if p is a member of E but not a member of I. $C \subset E$, $C \cup I = E$ and $C \cap I = \{\}$. Hence a *consequentially exposed* primary cell is only vulnerable to an external attacker when at least one other *exposed* primary cell has been exposed. When an external attacker has exposed a primary cell it was for one of two reasons, the cell was either initially or consequentially exposed. If $I = \{\}$ then both $C = \{\}$ and $E = \{\}$.

3 Proposition 2

Only the protection of the *initially exposed* primary cells, I, need to be considered when selecting secondary cells to suppress in order to make a published statistical table safe from an external attacker.

3.1 Proof

To protect the primary cells in a published statistical table a set of secondary cells, S, must be suppressed along with the primary (primary) cells, P. When

choosing S to protect the primary cells, P, minimising the loss of information from the published statistical table is considered. Let S_p be a set of secondary cells that protect p, $p \in P$. Let $L_{p \cup S_p}$ be the subset of linear equations M that contain the value $+1$ or -1 in the locations for a_p and all a_s where $s \in S_p$, $L_{p \cup S_p} \subseteq M$. This subset $L_{p \cup S_p}$ only contains the linear equations that apply to a_p and all associated a_s. The set of secondary cells, S_p, are primarily chosen so that $\underline{a_p} \leq lpl_p$ and $\overline{a_p} \geq upl_p$, where

$$\begin{aligned} \underline{a_p} &= min\ x_p \\ &s.t.\ L_{p \cup S_p} x = 0 \\ &\quad x_i \geq 0,\ i \in P \cup S_p \\ &\quad x_i = a_i, i \notin P \cup S_p \end{aligned} \qquad \text{and} \qquad \begin{aligned} \overline{a_p} &= max\ x_p \\ &s.t.\ L_{p \cup S_p} x = 0 \\ &\quad x_i \geq 0,\ i \in P \cup S_p \\ &\quad x_i = a_i, i \notin P \cup S_p \end{aligned}$$

From the definition of *initially exposed* primary cells we know that for $p \notin I$ that $\underline{a_p} \leq lpl_p$ and $\overline{a_p} \geq upl_p$ when $S_p = \{\}$. We also know that for $p \in I$ that $\underline{a_p} > lpl_p$ or $\overline{a_p} < upl_p$ when $S_p = \{\}$. From the definition of S_p we know that for $p \in I$ that $\underline{a_p} \leq lpl_p$ and $\overline{a_p} \geq upl_p$ when $S_p \neq \{\}$. Therefore the set of secondary suppressed cells, S_p, are only required to protect a_p when $p \in I$, they are not required in the protection of a_p when $p \notin I$. So, the only time $S_p \neq \{\}$ is when $p \in I$.

$$p \in I \Leftrightarrow S_p \neq \{\}$$

Hence only the protection of the *initially exposed* primary cells, I, need to be considered when selecting secondary cells to suppress in order to make a published statistical table safe from an external attacker.

3.2 Corollary

If $I = \{\}$ then $N = P$ and therefore the statistical table is already adequately protected.

4 Finding *initially exposed* primary cells without using a solver

We present here a method that provides a superset of the elements in P that contains all those in I.

For each element $p \in P$ let J denote the set of linear constraint equations (equivalent to rows of M) in which p participates, i.e. $\forall j \in J \cdot Mpj \neq 0$.

A necessary, but not sufficient, condition for us to establish that $p \in I$ is the existence of at least one marginal total in which the amount of "uncertainty" (and hence protection) provided by the absolute values of the other suppressed primary cells in that total is less than the required protection limits. Formally, for each $j \in J$ let H_j be the set of primary cells in j, we require that one of the following conditions holds:

$$|H_j| = 1 \quad \text{or}$$
$$|H_j| > 1 \quad \wedge \quad (Max(a_p - lpl_p, upl_p - a_p) > \sum_{i \in H_j/p} a_i)$$

4.1 Example

Taking a 6 by 6 statistical table with marginal totals (Table 1) as an example, the process of finding I, C and N can be shown. In our example the statistical agency has defined $P = \{8, 12, 15, 16, 19, 20, 24, 27\}$. When the test for the fully exposed primary cells is applied five primary cells are exposed, $E = \{16, 19, 20, 24, 27\}$ and therefore $N = \{8, 12, 15\}$. The values of cells 16, 20, 24 and 27 are calculated exactly and the feasibility range of cell 19 is calculated within its lower and upper protection levels which in this case is 10% of the cell's value.

By contrast applying the test for initially exposed primary cells (Table 2) we find that $I = \{16, 19, 24\}$, and therefore $N \cup C = \{8, 12, 15, 20, 27\}$. For this pre-processing optimisation to work it is not necessary (nor is it possible) to determine which cell is in C and which is in N.

	Total	1	2	3	4	5	6
Total	1472	193	278	203	294	233	271
A	199	[8] **9** [1]	51	41	47	[12] **3** [1]	48
B	164	[15] **8** [2]	[16] **1** [1]	54	44	[19] **45** [2]	[20] **12** [2]
C	245	8	70	[24] **6** [2]	76	64	[27] **21** [2]
D	248	33	46	45	27	37	60
E	312	87	51	18	35	49	72
F	304	48	59	39	65	35	58

Table 1. Example of a 6 by 6 statistical tables with marginal totals. There are 8 primary cells. Each primary cell has its cell number top left and number of contributors bottom right.

Applying a SAS/OR implementation of the classical IP SDC model to the whole set of primary cells in table 1 the set of secondary cells $S = \{37, 38, 40\}$ was obtained. The solver required 833 variables, 1824 constraints and 23.28 seconds of cpu time to protect table 1.

Applying a SAS/OR implementation of the modified classical IP SDC model to only the initially exposed primary cells, $I = \{16, 19, 24\}$, in table 1 the set of secondary cells $S = \{37, 38, 40\}$ was also obtained. The solver required 343 variables, 689 constraints and 3.75 seconds of cpu time to protect table 1.

Cell	Protection range	Sum of other Primary Cells in Row	Sum of other Primary Cells in Column	Protected
8	±1	3	8	Yes
12	±1	8	45	Yes
15	±1	58	9	Yes
16	±1	65	0	No
19	±4.5	21	3	No
20	±1.2	54	21	Yes
24	±1	21	0	No
27	±2.1	6	12	Yes

Table 2. Workings to find members of the superset of I. Any cell that has either a sum of other primary cells in either the row or column that is larger than it's protection range is a member of the superset of I.

5 Applying the Proposition to the Classical IP Model for SDC

The cell suppression problem is the problem faced by statistical agencies when they release statistical tables, they must balance the risk of disclosing confidential information against the loss of information from the table caused by not publishing the suppressed cells in the table [3] [7] [4] [8] [5].

Here we consider the case of a single external attacker who has no other knowledge than what is in the published table. It is usually assumed that the external attacker, prior to attack, knows that the cell a_i lies within the range from lb_i to ub_i. If the external attacker has no other knowledge than that published in the table then $lb_i = 0$ and $ub_i = \infty$. Fischetti and Salazar-González [3], when they defined the classical model, introduced a weighing w_i for each cell a_i to represent the information loss should the cell a_i be suppressed. A variable z_i was introduced for each a_i to indicate whether or not a_i had been suppressed ($z_i = 0$ means that a_i is published and $z_i = 1$ means that a_i is suppressed). Two tables where introduced that are consistent with $a = [a_1, ..., a_n]$, these tables $f^p = [f_1^p, ..., f_n^p]$ and $g^p = [g_1^p, ..., g_n^p]$ are used to calculate the lower and upper feasible limits for $p \in P$. In the classical model the lower and upper bounds (lb_i and ub_i) are translated into LB_i and UB_i, where $LB_i = a_i - lb_i$ and $UB_i = ub_i - a_i$. Those cells that are suppressed and are members of P are called primary suppressed cells and those cells that are suppressed but are not members of P are called secondary suppressed cells.

5.1 Classical Model

$$
\begin{aligned}
&min\ \textstyle\sum_{i=1}^{n} w_i z_i \\
&subject\ to \\
&\quad z_i \in \{0,1\} && for\ i = 1,...,n \\
&and\ for\ all\ p \in P: \\
&\quad \textstyle\sum_{i=1}^{n} M_{ij} f_i^p = 0 && for\ j = 1,...,m \\
&\quad a_i - LB_i z_i \le f_i^p \le a_i + UB_i z_i && for\ i = 1,...,n \\
&\quad \textstyle\sum_{i=1}^{n} M_{ij} g_i^p = 0 && for\ j = 1,...,m \\
&\quad a_i - LB_i z_i \le g_i^p \le a_i + UB_i z_i && for\ i = 1,...,n \\
&\quad f_p^p \le lpl_p \\
&\quad g_p^p \ge upl_p \\
&\quad g_p^p - f_p^p \ge spl_p
\end{aligned}
$$

5.2 Modified Classical Model

Applying propositions 1 and 2, the proof and the corollary in this paper we derived the Classic Model from Fischetti and Salazar as follows:

$$
\begin{aligned}
&min\ \textstyle\sum_{i=1}^{n} w_i z_i \\
&subject\ to \\
&\quad z_i \in \{0,1\} && for\ i = 1,...,n \\
&\quad z_p = 1 && for\ all\ p \in P \\
&and\ for\ all\ p \in I(initially\ exposed\ primary\ cells): \\
&\quad \textstyle\sum_{i=1}^{n} M_{ij} f_i^p = 0 && for\ j = 1,...,m \\
&\quad a_i - LB_i z_i \le f_i^p \le a_i + UB_i z_i && for\ i = 1,...,n \\
&\quad \textstyle\sum_{i=1}^{n} M_{ij} g_i^p = 0 && for\ j = 1,...,m \\
&\quad a_i - LB_i z_i \le g_i^p \le a_i + UB_i z_i && for\ i = 1,...,n \\
&\quad f_p^p \le lpl_p \\
&\quad g_p^p \ge upl_p \\
&\quad g_p^p - f_p^p \ge spl_p
\end{aligned}
$$

6 Experimental Setup

6.1 Comparing the Classical and Modified Classical Models

A set of 20 2-dimensional non-hierarchical magnitude statistical tables with marginal totals (see Table 3) were generated for the purpose of comparing the

classical and modified models [8]. These statistical tables with marginal totals were protected using a SAS/OR implementation of the classical model and a SAS/OR implementation of the modified (initially exposed primary cells only) classical model, using the same computer. These experiments were ran at ONS on a Dell Optiplex GX270 processor with 2GB RAM. The SAS version used was SAS 9 solver with SAS/OR Opt module. There are a variety of solvers in SAS and OptMILP was used. The selected secondary suppressed cells, the number of variables required, the number of constraints and the required cpu-time were recorded for comparison. For each of the statistical tables the percentage change in performance was calculated using the following formula.

$$ReductionInCellsConsidered = \frac{(SensitiveCells - InitiallyExposedCells) * 100}{SensitiveCells}$$

$$ImprovementInVariables = \frac{(ClassicalVariables - ModifiedVariables) * 100}{ClassicalVariables}$$

$$ImprovementInConstraints = \frac{(ClassicalConstraints - ModifiedConstraints) * 100}{ClassicalConstraints}$$

$$ImprovementInCPUTime = \frac{(ClassicalCPUTime - ModifiedCPUTime) * 100}{ClassicalCPUTime}$$

For each of these statistical tables the improvement in the number of variables, constraints and cpu time was plotted against the reduction in the number of primary cells needing to be considered, see Fig. 1.

6.2 Estimating the Improvement for Different Table Sizes

A set of 3360 2-dimensional non-hierarchical statistical tables with marginal totals, sizes ranging from 100 cells to 900,000 cells, were generated with random values. For each different table size; 40 tables were generated, these tables had either 10% or 25% primary cells and either 10% or 20% of cells set to zero. For each of these tables the percentage reduction in the number of primary cells that need to be considered when using the modified classical model was plotted against the table size, see Fig. 2.

7 Results

7.1 Comparing the Classical and Modified Classical Models

Both models, classical and modified, selected the same secondary cells to suppress. The number of variables required, the number of constraints and the required cpu-time for each model is recorded in Table 4.

Table	Rows	Columns	Cells	Zeros	Primary Cells	Initially Exposed	Constraint Equations	Hierarchical
1	5	5	36	3	8	6	12	No
2	5	6	42	7	8	3	13	No
3	5	7	48	5	7	4	14	No
4	5	8	54	8	17	3	15	No
5	5	9	60	5	17	4	16	No
6	7	7	64	11	14	5	16	No
7	7	8	72	7	16	5	17	No
8	7	9	80	19	13	5	18	No
9	8	8	81	13	13	7	18	No
10	8	9	90	15	17	5	19	No
11	10	10	121	19	31	4	22	No
12	10	12	143	28	40	4	24	No
13	25	5	156	5	4	4	32	No
14	25	5	156	6	11	8	32	No
15	25	5	156	7	4	4	32	No
16	25	5	156	32	7	7	32	No
17	25	5	156	35	7	4	32	No
18	25	5	156	26	9	9	32	No
19	25	5	156	7	11	10	32	No
20	50	5	300	9	25	18	56	No

Table 3. Range of statistical tables with marginal totals

For every percentage reduction in the number of primary cells that need to be considered when using the modified classical model to protect a published statistical table there is an equal percentage improvement in the number of variables and constraints required to solve the associated linear programme. There is also a similar improvement in the required cpu time, however the relationship is not as smooth as it is for the number variables and constraints required, see Fig. 1. For those statistical tables where all of the primary cells are initially exposed, $P = I$, the modified classical model may require more cpu time than the classical model.

7.2 Estimating the Improvement for Different Table Sizes

The reduction in the number of primary cells that needed to be considered when using the modified classical model was affected by some of the properties of the statistical tables being protected. The reduction was greater for larger tables, tables that were more square than long and tables that had a higher proportion of primary cells. This is explained by each factor increasing the probability that more than one primary cell would occupy the same row or column and hence provide some protection to each other.

Table	Classical			Modified		
	Variables	Constraints	cpu-time	Variables	Constraints	cpu-time
1	612	1376	4.32	468	1034	2.95
2	714	1584	3.71	294	599	1.32
3	720	1568	8.17	432	899	2.43
4	1890	4250	4.07	378	764	0.6
5	2100	4692	4.17	540	1117	0.98
6	1856	4088	8.31	704	1469	2.39
7	2376	5216	31.07	792	1641	4.62
8	2160	4680	113.78	880	1808	27.48
9	2187	4732	38.23	1215	2554	24.65
10	3150	6834	84.81	990	2022	6.98
11	7623	16492	95.56	1089	2155	2.57
12	11583	24960	256.65	1287	2532	5.98
13	1404	2768	4.82	1404	2768	4.86
14	3588	7612	78.46	2652	5539	62.9
15	1404	2768	31.57	1404	2768	31.7
16	2340	4844	18.07	2340	4844	29.82
17	2340	4844	22.67	1404	2771	8.45
18	2964	6228	267.31	2964	6228	267.31
19	3276	6921	2.23	3276	6921	2.2
20	15300	32900	110.70	11100	23695	45.96

Table 4. Comparison of the two models

8 Conclusions

This pre-processing optimisation has been shown to be very effective when applied to the classical IP SDC model developed by Fischetti and Salazar-González [3]. This optimisation works by reducing the resources that the solver requires to protect statistical tables, hence allowing statistical tables to be protected quicker or allowing larger statistical tables to be protected. The classical IP SDC model has been implemented, as the Optimal Method, in the SDC tool, τ-Argus [5] [9]. It may be the case that this pre-processing optimisation could be applied to the τ-Argus Optimal Method to enable it to handle larger tables.

9 Further Research

How the properties of the statistical tables affect the amount of improvement that this pre-processing optimisation provides requires further investigation. How hierarchical statistical tables affect the amount of improvement that this pre-processing optimisation provides requires further investigation. This pre-processing optimisation should be applied to other SDC techniques to see if similar performance improvements can be obtained.

Table	Reduction in Cells Considered	Improvement in Variables	Improvement in Constraints	Improvement in CPU Time
1	25	23.52	24.85	31.71
2	62.5	58.82	62.18	64.4
3	42.86	40	42.67	70.26
4	82.35	80	82.02	85.26
5	76.47	74.29	76.19	76.5
6	64.29	62.07	64.07	71.24
7	68.75	66.67	68.54	85.13
8	61.54	59.26	61.37	75.85
9	46.15	44.44	46.03	35.52
10	70.59	68.57	70.41	91.77
11	87.1	85.71	86.93	97.31
12	90	88.89	89.86	97.67
13	0	0	0	-0.83
14	27.27	26.09	27.23	19.83
15	0	0	0	-0.41
16	0	0	0	-65.02
17	42.86	40	42.80	62.73
18	0	0	0	0
19	9.09	0	0	1.35
20	28	27.45	27.98	58.48

Table 5. Percentage Reduction in Primary Cells Considered, the Number of Variables needed by SAS/OR, the Number of Constraints needed by SAS/OR and the CPU Time needed by SAS/OR

References

1. Castro, J.: A shortest paths heuristic for statistical data protection in positive tables. Research Report DR 2004-10. Report available from http://www-eio.upc.es/j̃castro. (2005)
2. Clark, A. and J. Smith: Improvements to Cell Suppression in Statistical Disclosure Control. End-of-Project Report ONS Contract IT-06-0960A for the Office of National Statistics (ONS), (2006)
3. Fischetti, M. and J.J. Salazar-González: Solving the Cell Suppression Problem on Tabular Data with Linear Constraints. Management Science 2001 INFORMS. Vol. 47, No. 7, July 2001 pp. 1008-1027 (2001)
4. Giessing, S: Handbook on Statistical Disclosure Control, Version 1.01, Ch 4. CENEX SDC, a CENtre of EXcellence for Statistical Disclosure Control (2007)
5. Hundepool, A., A. van de Wetering, R. Ramaswamy, P. Wolf, S. Giessing, M. Fischetti, J.J. Salazar, J. Castro and P. Lowthian: Tau-ARGUS Users Manual. CENEX-project. BPA no: 769-02-TMO (2007)
6. Salazar-González, J.J.: Extending Cell Suppression to Protect Tabular Data against Several Attackers. J. Domingo-Ferrer (Ed.): Inference Control in Statistical Databases, LNCS 2316, pp. 34-58, 2002. (2002)

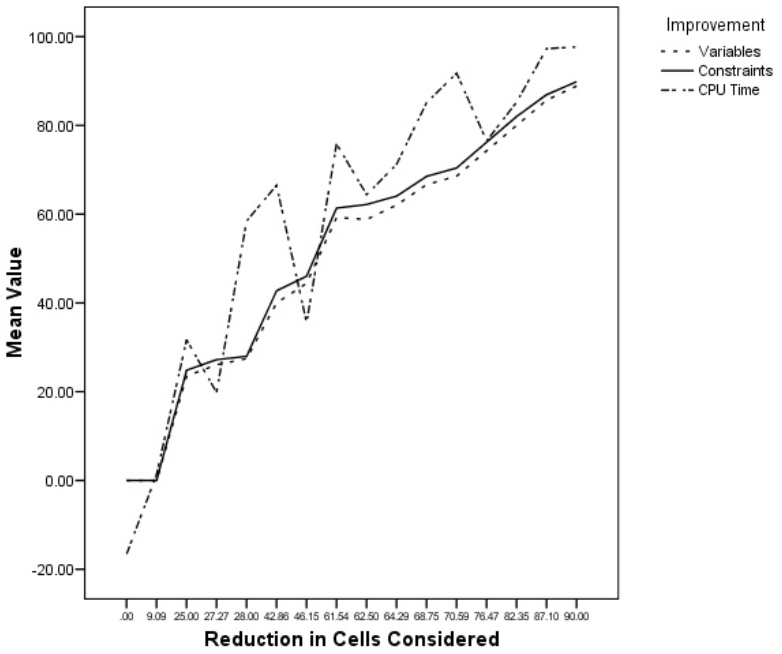

Fig. 1. Percentage Improvement in Number of Variables needed by SAS/OR, the Number of Constraints needed by SAS/OR and the CPU Time needed by SAS/OR by the Percentage Reduction in Primary Cells Considered.

7. Shlomo, N. and C. Young: Quality Measures for Statistical Disclosure Controlled Data. Proceedings of the European Conference on Quality in Survey Statistics (2006)
8. Staggemeier, A.T., A.R. Clark, J. Smith, and J. Thompson: Improving our knowledge of metaheuristic approaches for cell suppression problem. Joint UN-ECE/Eurostat work session on statistical data confidentiality, Manchester, United Kingdom, 17-19 December (2007)
9. Willenborg, L. and T. de Waal.: Elements of Statistical Disclosure Control. New York: Springer. (2001)

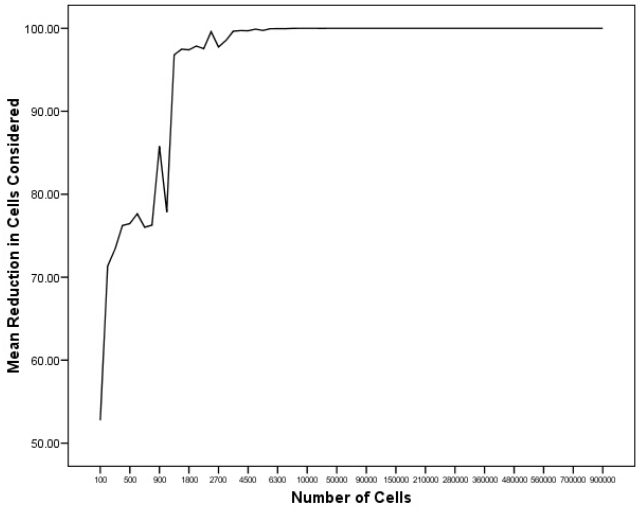

Fig. 2. The Percentage Reduction in Primary Cells Considered by the Number of Cells in the Table.